CRICKET WORLD CUP 2019 BOWLERS PERFORMANCE ANALYSIS

By : Partheesh Ranjan Singh PES2201800060 Section - D

Nishanth S Shastry PES2201800533 Section - A

INTRODUCTION

Our analysis basically shows how bowlers who were in the tournament performed according to attributes such as Economy, Average, Ground, Overs, etc.

WE WILL BE USING DIFFERENT CHARTS TO SHOWCASE OUR ANALYSIS WHICH WILL BE USER UNDERSTANDABLE AND WILL GIVE THE RESULTS OF THE PERFORMANCE OF VARIOUS BOWLERS WHO WERE PART OF THEIR TEAMS.

TIME

3 WEEKS

OBJECTIVES

- The objective of this project is to use the salient features of R to represent analysis of the performance of the players(bowlers).
- To provide a nice and colourful user interface to represent our visualization with the necessary visualization methods.
- To use different charts to showcase our analysis which will be user understandable and will give the results of the performance of various bowlers who were part of their teams.
- To find out how bowlers performed in various conditions and different grounds and hence sought out to analyze the performance of bowlers across the globe.

DATASET DETAILS

- **Title:** - Cricket World Cup 2019 Bowlers Performance Analysis
- **Dataset Name:** - Bowler_data.csv
- **Dataset Source:** - Bowler Dataset(Internet link)
- **Description:** - The dataset is created by 'SaiVamshiAtukuri' (owner) to analyze the performance of all the players playing in CWC 2019.The data is scraped from ESPN Cricinfo .Matches till 18th May 2019 are only counted. Match level Bowlers data of all cricketers playing in CWC2019.
- **Number of rows:** - 11118
- **Number of columns:** - 14

VARIABLE DESCRIPTION

1. Overs : - The no. of overs played by each player in the dataset.
2. Mdns : - The no. of maiden overs bowled(all 6 balls score is 0).
3. Runs : - The runs scored by the player.
4. Wkts : - The no. of wickets taken by the player.
5. Econ : - The average number of runs conceded for each over bowled.
6. Ave : - Player's batting average is the total number of runs they have scored divided by the number of times they have been out.

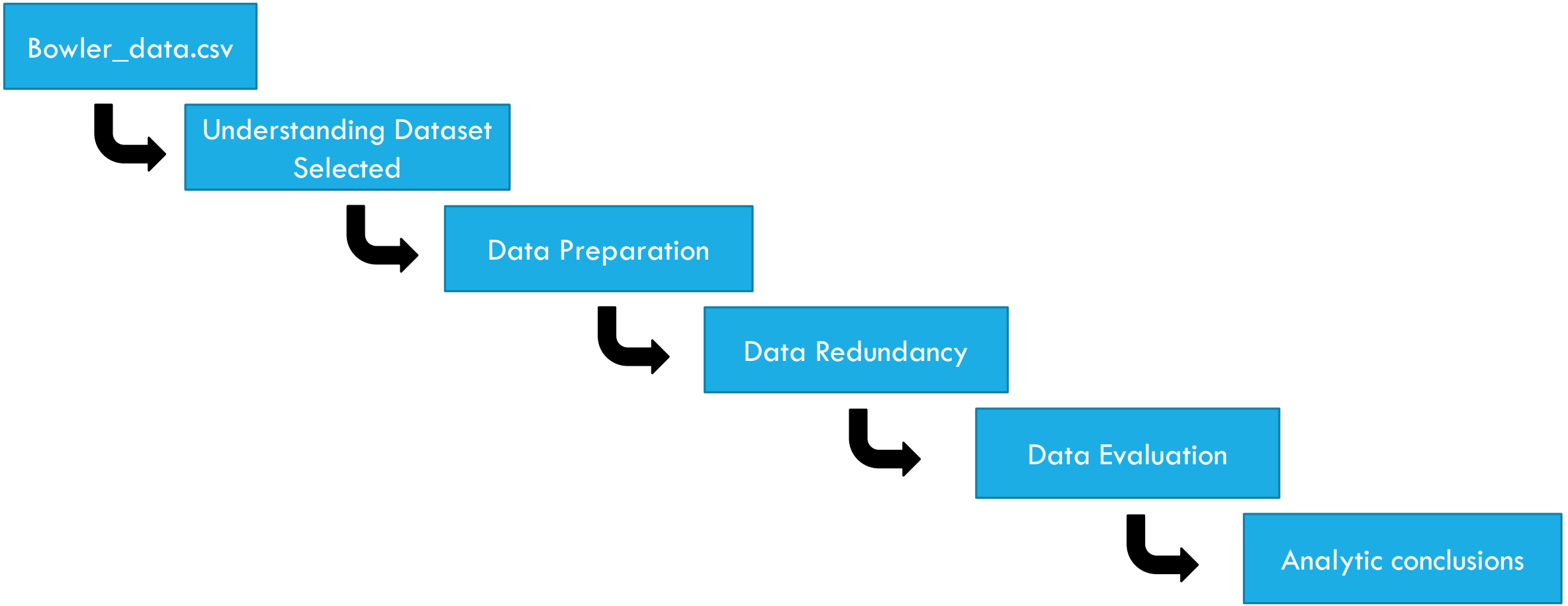
VARIABLE DESCRIPTION (CONTD…)

7. SR : - The measure of how quickly a batsman achieves the primary goal of batting, namely scoring runs.
8. Opposition : - The team against which the player's team played against.
9. Ground : - The location/place where the player played the game.
10. Start Date : - The date on which the game took place.
11. Match_ID : - Every match played has a unique ID.
12. Bowler : - The player/bowler name.
13. Player_ID : - Every player/bowler has a unique ID.

APPROACH

- Used the data set from Kaggle.com.
- Applied the learnings from class modules and online resources.
- Applied basic principles of data cleansing and preparation.
- Made some basic assumptions.
- Analyzed data with various plots.

METHODOLOGY



LITERATURE SURVEY - PARTHEESH RANJAN SINGH

CricAI A classification based tool to predict the outcome in ODI cricket.pdf

1.Title: CricAI: A classification based tool to predict the outcome in ODI cricket

Publisher: IEEE

2. AUTHORS: Amal Chaminda Kaluarachchi , Aparna S. Varde

3. Published in : 2010 Fifth International Conference on Information and Automation for Sustainability of IEEE.

4.Survey link: - LITERATURE SURVEY - Partheesh Ranjan Singh .pdf

LITERATURE SURVEY – NISHANTH S SHASTRY

Prediction of Cricket World Cup 2019 by TOPSIS Technique of MCDM-A Mathematical Analysis.pdf

1. **Title: -** Prediction of Cricket World Cup 2019 by TOPSIS Technique of MCDM-A Mathematical Analysis
2. **Publisher: -** Google Scholar
3. **Authors: -** Muhammad Saqlain, Naveed Jafar, Rashid Hamid, Amir Shahzad
4. **Published in: -** International Journal of Scientific & Engineering Research Volume 10, Issue 2, February-2019 ISSN 2229-5518 IJSER © 2019
5. **Survey link: - Literature Survey_Nishanth_Shastry.pdf**

DATA ANALYSIS

Following Analysis will be the part of project: -

1. Overs vs Runs
2. Overs vs Wickets
3. Overs vs Maidens
4. Overs vs Economy
5. Overs vs Average
6. Bowlers and Economy
7. Average vs Ground
8. Opposition vs Number of wickets
9. Bowler's 5 wickets vs opposition
10. Runs vs Wickets
11. Bowlers vs Wickets

ANALYSIS
OF
SELECTED DATASET

‘Bowler_data.csv’

PROFILING THE DATA

Code: -

```
1  #Profiling the data
2
3  MyFile <-"Bowler_data.csv"
4  MyData <- read.csv(file=MyFile, header=TRUE, sep=",")
5  options(max.print = 12000)
6  print(dim(MyData))                          #Excluding the header names of columns - 11118 14
7
```

Output: -

```
> MyFile <-"Bowler_data.csv"
> MyData <- read.csv(file=MyFile, header=TRUE, sep=",")
> options(max.print = 12000)
> print(dim(MyData))                          #Excluding the header names of columns - 11118 14
[1] 11118    14
>
```

DATA CLEANING

```
> head(MyData)
  X Overs Mdns Runs Wkts Econ  Ave SR   Opposition  Ground Start.Date   Match_ID         Bowler Player_ID
1 1     8    0   57    0 7.12    -  -      v India  Nagpur  18-Dec-09 ODI # 2933 Suranga Lakmal     49619
2 2    10    0   55    2  5.5 27.5 30     v India Kolkata  24-Dec-09 ODI # 2935 Suranga Lakmal     49619
3 3     -    -    -    -    -    -  -     v India   Delhi  27-Dec-09 ODI # 2936 Suranga Lakmal     49619
4 4     9    1   63    2    7 31.5 27 v Bangladesh   Dhaka   4-Jan-10 ODI # 2937 Suranga Lakmal     49619
5 5     8    1   48    0    6    -  -     v India   Dhaka   5-Jan-10 ODI # 2938 Suranga Lakmal     49619
6 6    10    0   75    0  7.5    -  -     v India   Dhaka  10-Jan-10 ODI # 2941 Suranga Lakmal     49619
>
```

```
> mean(is.na(MyData))
[1] 6.424588e-06
>
```

```
 8  #Data Cleaning
 9  #Replacing '-' with values with respective to the columns
10
11  library(car)
12  #This library helps to replace the content element of csv file with required element
13  #We can use recode function for the changes
14
```

Replacing ‘-’ with ‘0’ in ‘Bowler_data.csv’ to Sort: -

Code: -

```
15 MyData$Overs <- recode(MyData$Overs,"c('-')='0'")
16 MyData$Mdns <- recode(MyData$Mdns,"c('-')='0'")
17 MyData$Wkts <- recode(MyData$Wkts,"c('-')='0'")
18 MyData$SR <- recode(MyData$SR,"c('-')='0'")
19 MyData$Runs <- recode(MyData$Runs,"c('-')='0'")
20 MyData$Ave <- recode(MyData$Ave,"c('-')='0.00'")
21 MyData$Econ <- recode(MyData$Econ,"c('-')='0.00'")
22
23 #Make all the changes and store the cleaned dataset in new .csv file
24 write.csv(MyData, file = "Bowlerdata_cleaned.csv")
25
26 #Check if dataset has the above changes
27 print(head(MyData))
28
```

Dataset in ‘Bowlerdata_cleaned.csv’: -

```
> print(head(MyData))
  X Overs Mdns Runs Wkts Econ  Ave SR   Opposition  Ground Start.Date   Match_ID         Bowler Player_ID
1 1     8    0   57    0 7.12 0.00  0      v India  Nagpur  18-Dec-09 ODI # 2933 Suranga Lakmal     49619
2 2    10    0   55    2  5.5 27.5 30      v India Kolkata  24-Dec-09 ODI # 2935 Suranga Lakmal     49619
3 3     0    0    0    0 0.00 0.00  0      v India   Delhi  27-Dec-09 ODI # 2936 Suranga Lakmal     49619
4 4     9    1   63    2    7 31.5 27 v Bangladesh   Dhaka   4-Jan-10 ODI # 2937 Suranga Lakmal     49619
5 5     8    1   48    0    6 0.00  0      v India   Dhaka   5-Jan-10 ODI # 2938 Suranga Lakmal     49619
6 6    10    0   75    0  7.5 0.00  0      v India   Dhaka  10-Jan-10 ODI # 2941 Suranga Lakmal     49619
> 
```

Dataset in ‘Bowler_data.csv’: -

```
> head(MyData)
  X Overs Mdns Runs Wkts Econ  Ave SR   Opposition  Ground Start.Date   Match_ID         Bowler Player_ID
1 1     8    0   57    0 7.12    -  -      v India  Nagpur  18-Dec-09 ODI # 2933 Suranga Lakmal     49619
2 2    10    0   55    2  5.5 27.5 30      v India Kolkata  24-Dec-09 ODI # 2935 Suranga Lakmal     49619
3 3     -    -    -    -    -    -  -      v India   Delhi  27-Dec-09 ODI # 2936 Suranga Lakmal     49619
4 4     9    1   63    2    7 31.5 27 v Bangladesh   Dhaka   4-Jan-10 ODI # 2937 Suranga Lakmal     49619
5 5     8    1   48    0    6    -  -      v India   Dhaka   5-Jan-10 ODI # 2938 Suranga Lakmal     49619
6 6    10    0   75    0  7.5    -  -      v India   Dhaka  10-Jan-10 ODI # 2941 Suranga Lakmal     49619
> 
```

Sorting Data to perform Data Visualization: -

Code: -

```
29  Data<-read.csv("Bowlerdata_cleaned.csv")
30  library(plyr)
31  #The plyr library has an inbuilt function called arrange which helps to sort
32  #the dataset in which ever order required(sorting based on variable specified).
33  # Sort by Overs
34  df <- Data
35  df_sorted <- arrange(df, Overs)
36  #check if the data has been sorted
37  head(df_sorted)
38  write.csv(df_sorted, file = "Bowlerdata_cleaned_sorted.csv")
39
```

Dataset in ‘Bowlerdata_cleaned_sorted.csv’: -

```
> head(df_sorted)
  X.1   X Overs Mdns Runs Wkts Econ Ave SR   Opposition        Ground Start.Date   Match_ID          Bowler Player_ID
1   3   3     0    0    0    0    0   0  0      v India         Delhi  27-Dec-09 ODI # 2936  Suranga Lakmal     49619
2  60  60     0    0    0    0    0   0  0 v Bangladesh      Dambulla  28-Mar-17 ODI # 3856  Suranga Lakmal     49619
3  88  88     0    0    0    0    0   0  0    v England   St George's  25-Feb-19 ODI # 4098   Oshane Thomas    914567
4 101 101     0    0    0    0    0   0  0 v Bangladesh    Chattogram  18-Oct-11 ODI # 3202   Andre Russell    276298
5 156 156     0    0    0    0    0   0  0  v Sri Lanka Colombo (SSC)  31-Jan-11 ODI # 3092     Kemar Roach    230553
6 228 228     0    0    0    0    0   0  0    v England    Bridgetown  20-Feb-19 ODI # 4096 Nicholas Pooran    604302
>
```

Dataset in ‘Bowler_data.csv’: -

```
> head(MyData)
  X Overs Mdns Runs Wkts Econ  Ave SR   Opposition  Ground Start.Date   Match_ID          Bowler Player_ID
1 1     8    0   57    0 7.12    -  -      v India  Nagpur  18-Dec-09 ODI # 2933  Suranga Lakmal     49619
2 2    10    0   55    2  5.5 27.5 30      v India Kolkata  24-Dec-09 ODI # 2935  Suranga Lakmal     49619
3 3     -    -    -    -    -    -  -      v India   Delhi  27-Dec-09 ODI # 2936  Suranga Lakmal     49619
4 4     9    1   63    2    7 31.5 27 v Bangladesh   Dhaka   4-Jan-10 ODI # 2937  Suranga Lakmal     49619
5 5     8    1   48    0    6    -  -      v India   Dhaka   5-Jan-10 ODI # 2938  Suranga Lakmal     49619
6 6    10    0   75    0  7.5    -  -      v India   Dhaka  10-Jan-10 ODI # 2941  Suranga Lakmal     49619
>
```

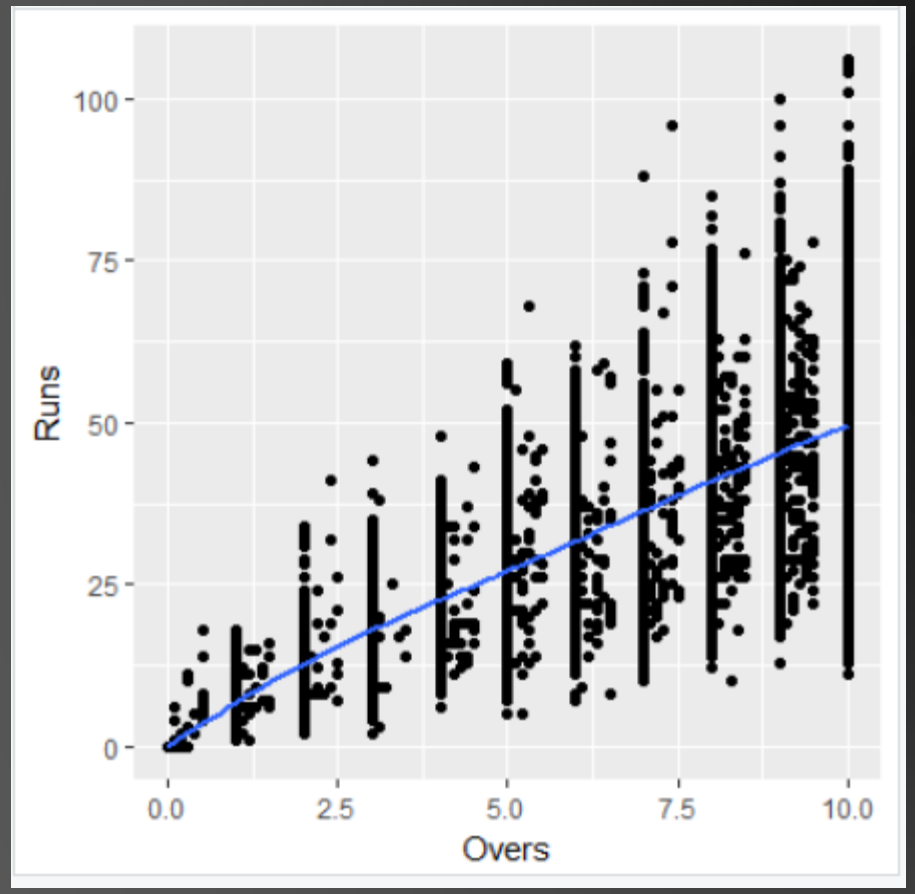
DATA VISUALIZATION

```
40  #Data Visualization
41
42  library(ggplot2)
43
```

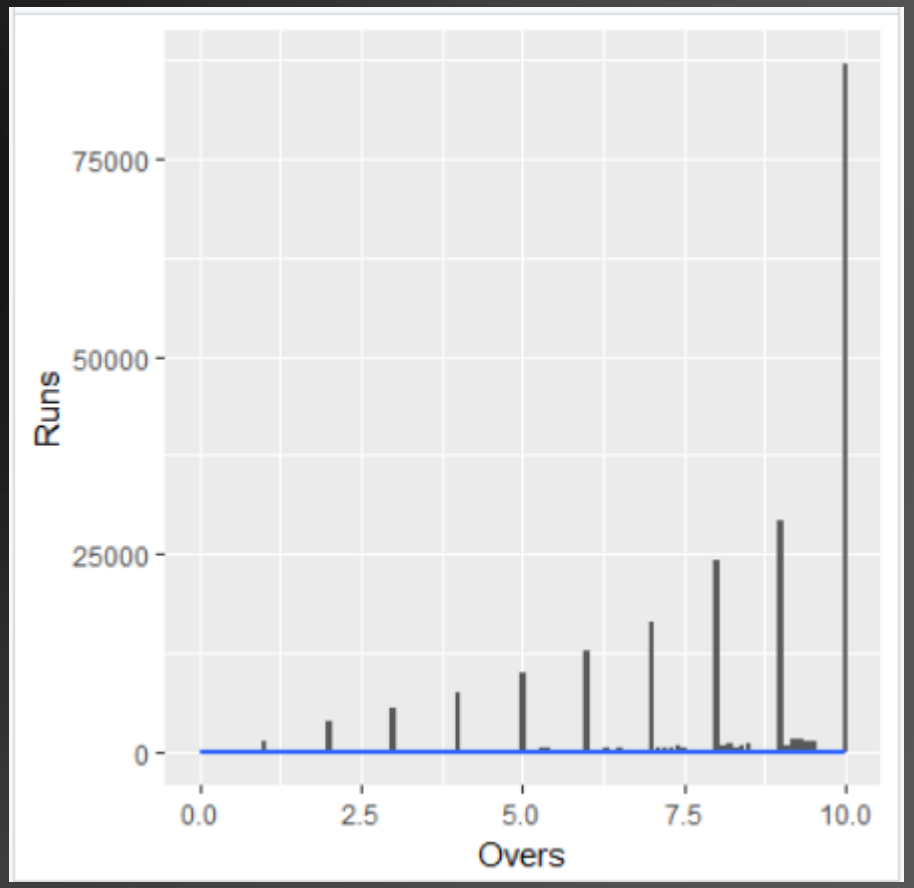
1. Overs vs Runs
2. Overs vs Wickets
3. Overs vs Maidens
4. Overs vs Economy
5. Overs vs Average
6. Bowlers and Economy
7. Average vs Ground
8. Bowler's 5 wickets vs Opposition
9. Opposition vs no. of wickets
10. Runs vs Wickets
11. Bowlers vs Wickets

1.Overs vs Runs

```
44  a<-ggplot(df_sorted,aes(x=Overs,y=Runs))
45  print(a+geom_point()+geom_smooth(method="loess",se=F))
46
```

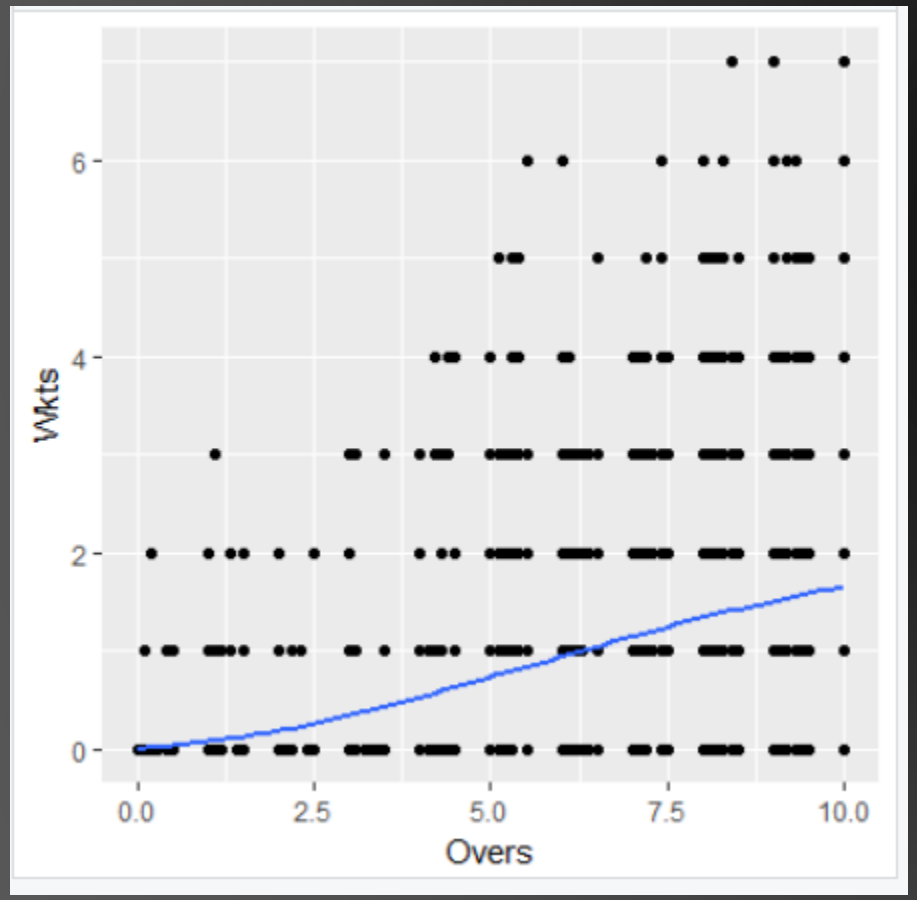



```
47  a<-ggplot(df_sorted,aes(x=Overs,y=Runs))
48  print(a+geom_col()+geom_smooth(method="loess",se=F))
49
```

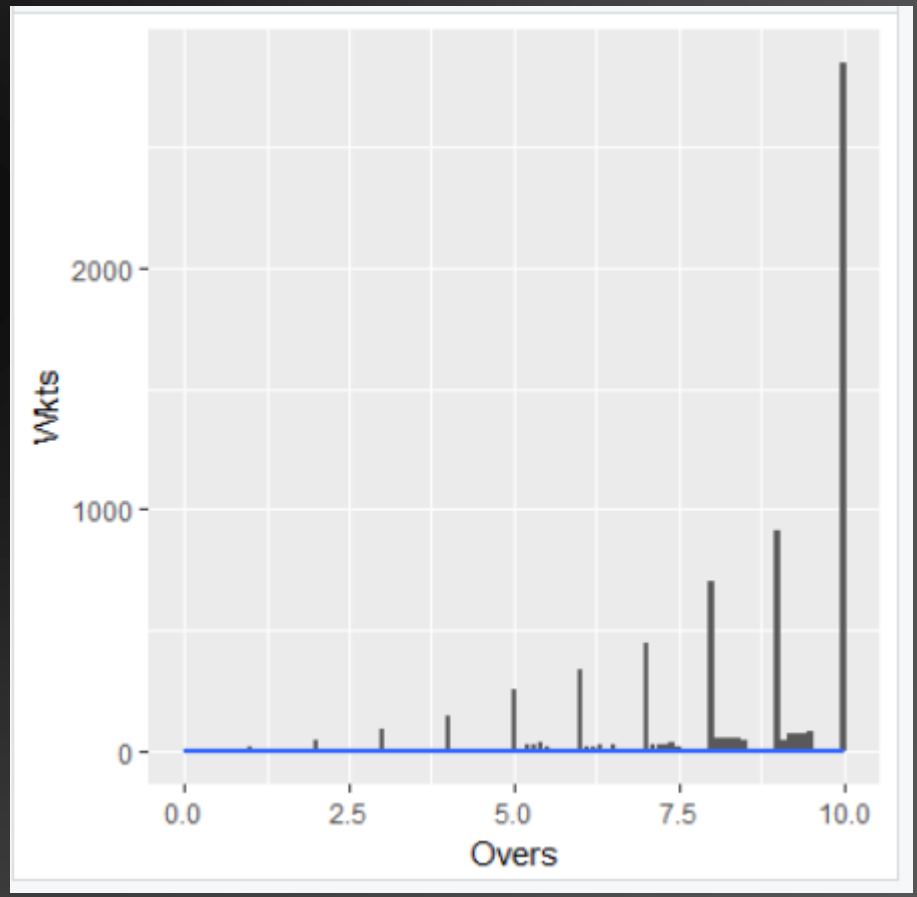


2. Overs vs Wickets

```
50 b<-ggplot(df_sorted,aes(x=Overs,y=Wkts))
51 print(b+geom_point()+geom_smooth(method="loess",se=F))
52
```

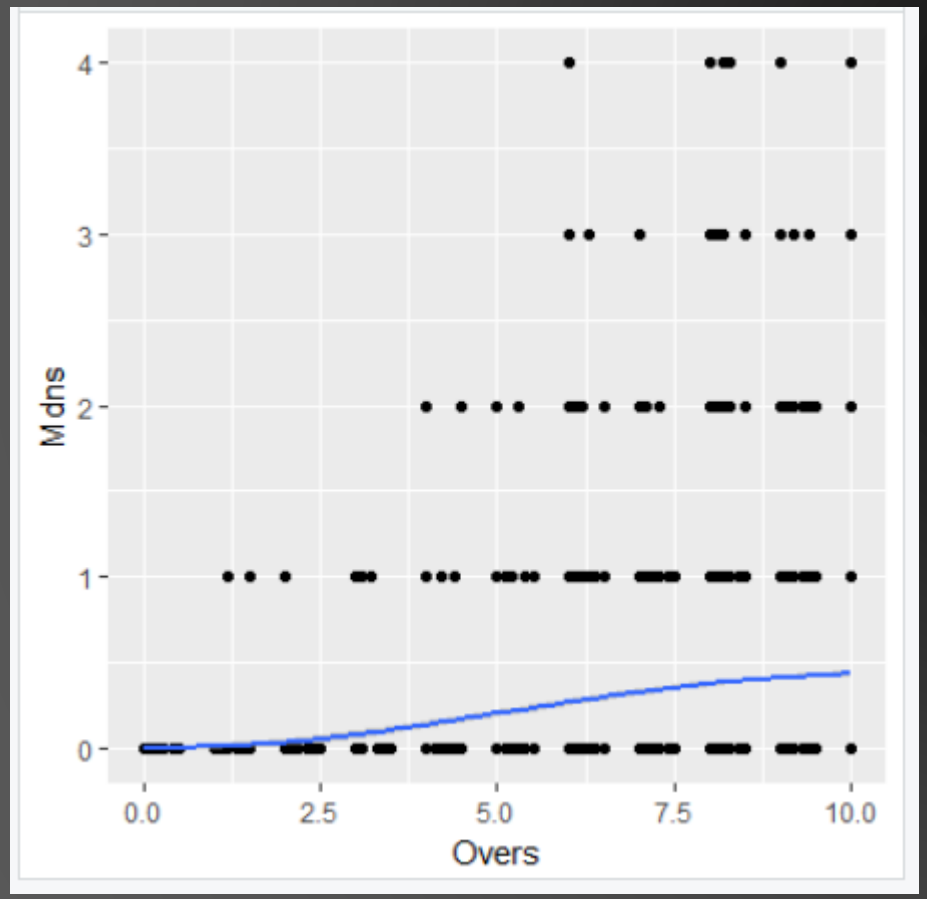



```
53 b<-ggplot(df_sorted,aes(x=Overs,y=Wkts))
54 print(b+geom_col()+geom_smooth(method="loess",se=F))
55
```

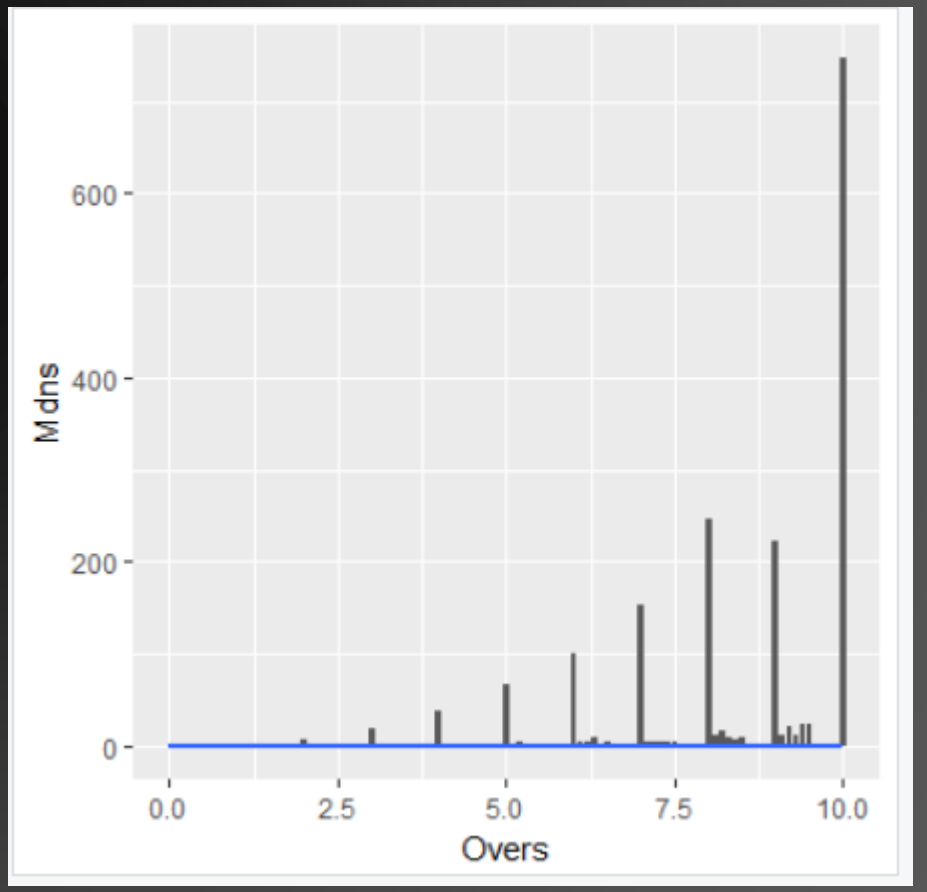


3. Overs vs Maidens

```
56  c<-ggplot(df_sorted,aes(x=Overs,y=Mdns))
57  print(c+geom_point()+geom_smooth(method="loess",se=T))
58  # output for lines 57
59  # Warning messages:
60  # 1: Removed 1 rows containing non-finite values (stat_smooth).
61  # 2: Removed 1 rows containing missing values (geom_point).
62
```

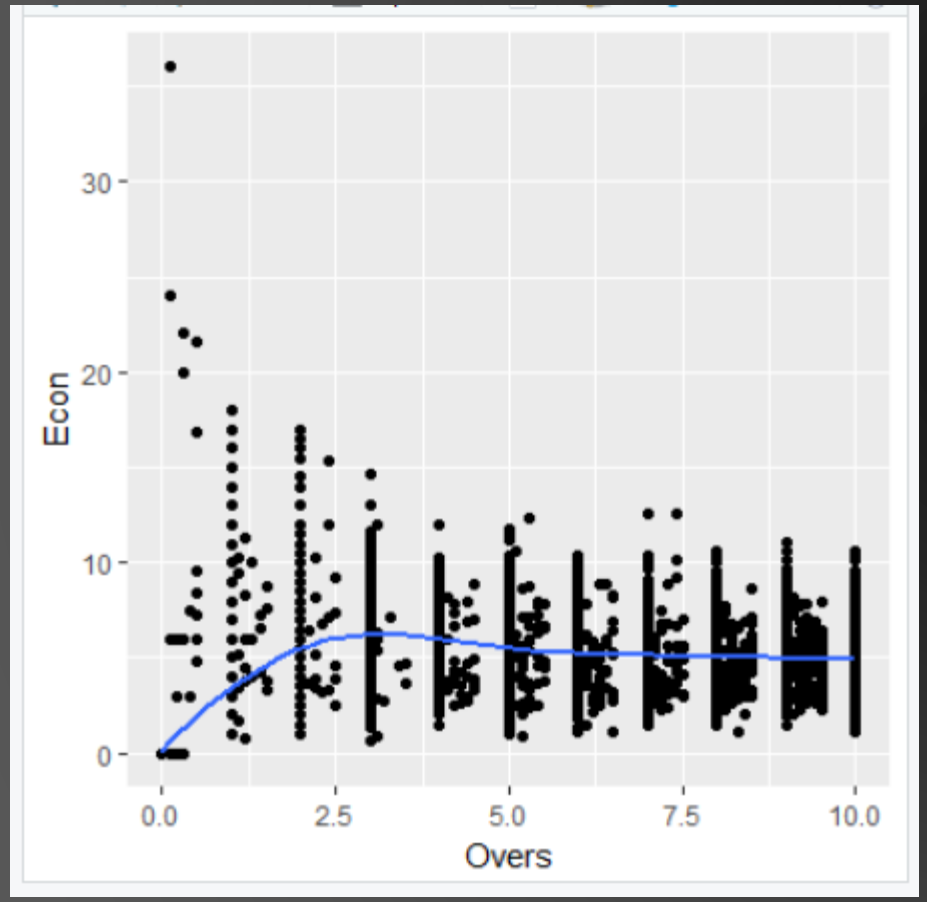



```
63  c<-ggplot(df_sorted,aes(x=Overs,y=Mdns))
64  print(c+geom_col()+geom_smooth(method="loess",se=T))
65  # output for lines 64
66  # Warning messages:
67  # 1: Removed 1 rows containing non-finite values (stat_smooth).
68  # 2: Removed 1 rows containing missing values (position_stack).
69
```

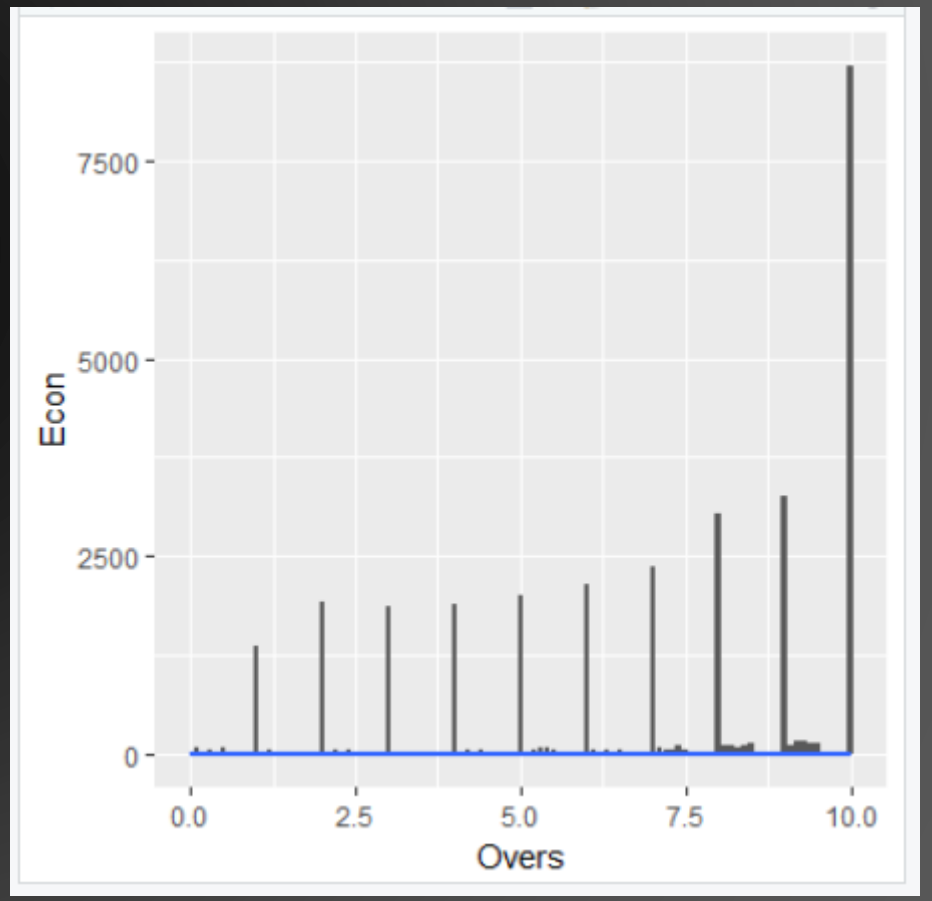


4. Overs vs Economy

```
70  d<-ggplot(df_sorted,aes(x=Overs,y=Econ))
71  print(d+geom_point()+geom_smooth(method="loess",se=T))
72
```

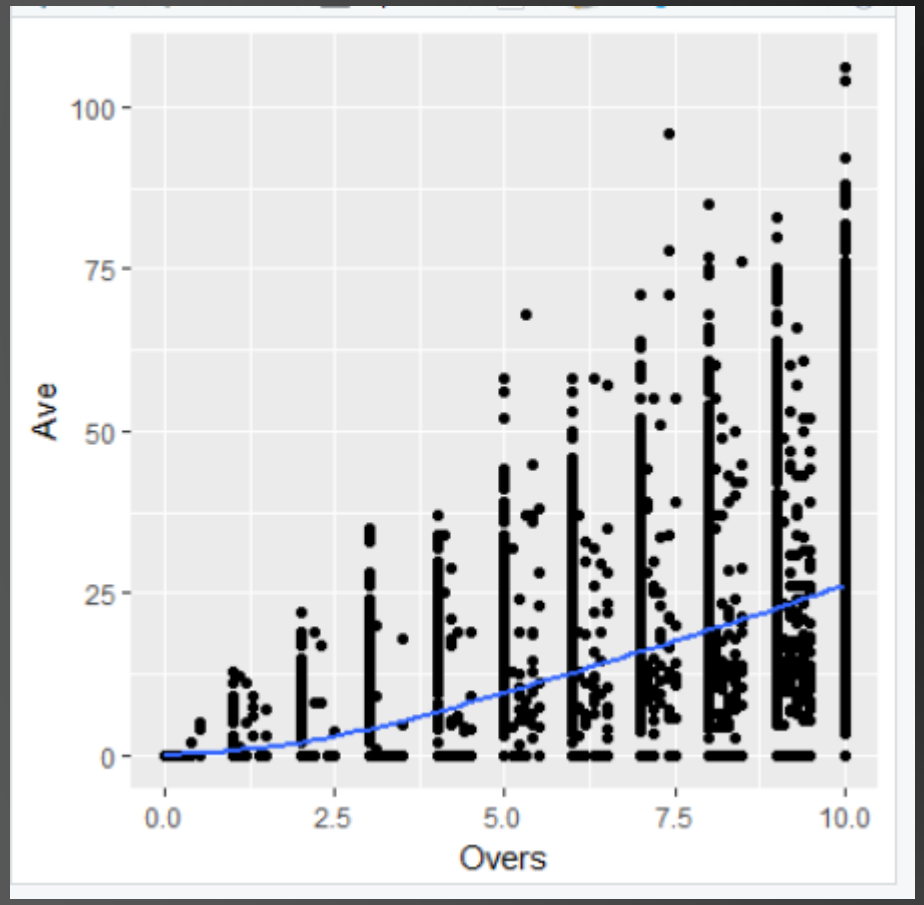



```
73  d<-ggplot(df_sorted,aes(x=Overs,y=Econ))
74  print(d+geom_col()+geom_smooth(method="loess",se=T))
75
```

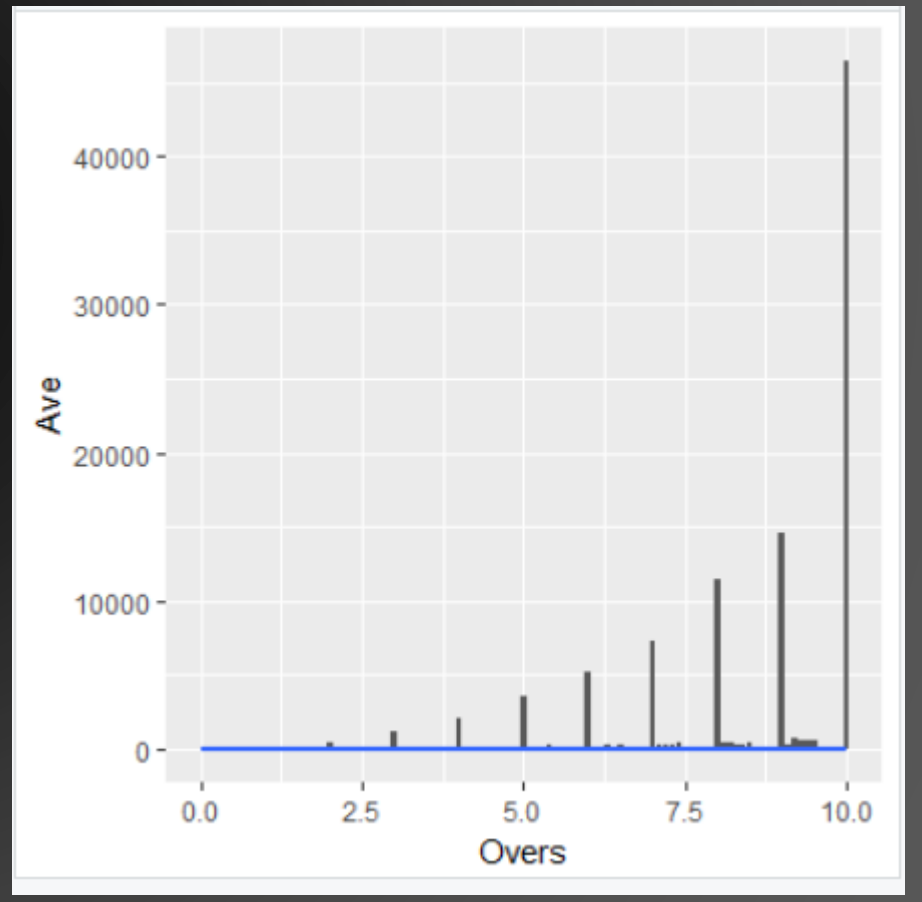


5. Overs vs Average

```
82  f<-ggplot(df_sorted,aes(x=Overs,y=Ave))
83  print(f+geom_point()+geom_smooth(method="loess",se=F))
84
```

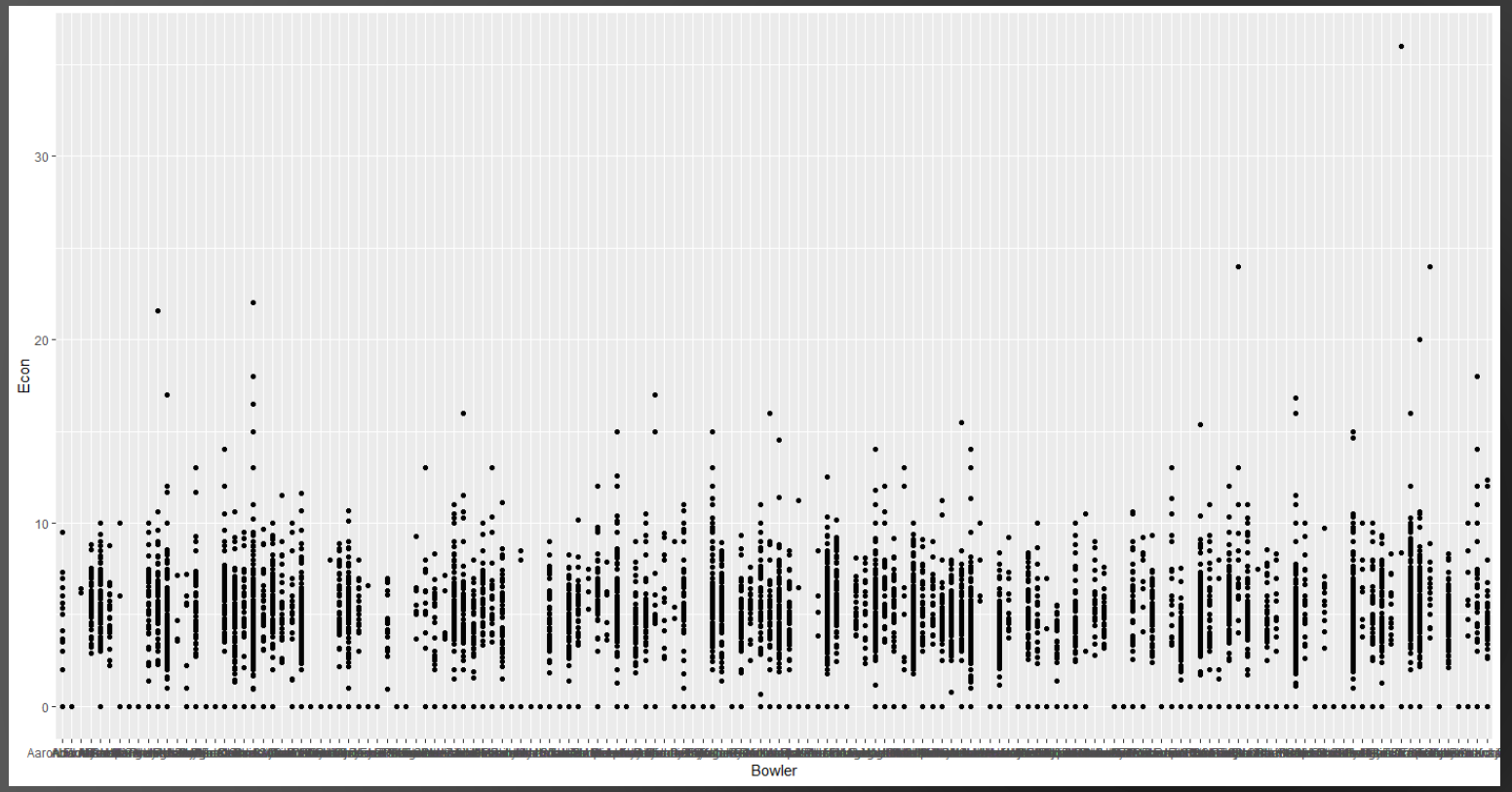



```
85  f<-ggplot(df_sorted,aes(x=Overs,y=Ave))
86  print(f+geom_col()+geom_smooth(method="loess",se=F))
87
```



6. Bowlers vs Economy

```
76  e<-ggplot(df_sorted,aes(x=Bowler,y=Econ))
77  print(e+geom_point()+geom_smooth(method="loess",se=F))
```



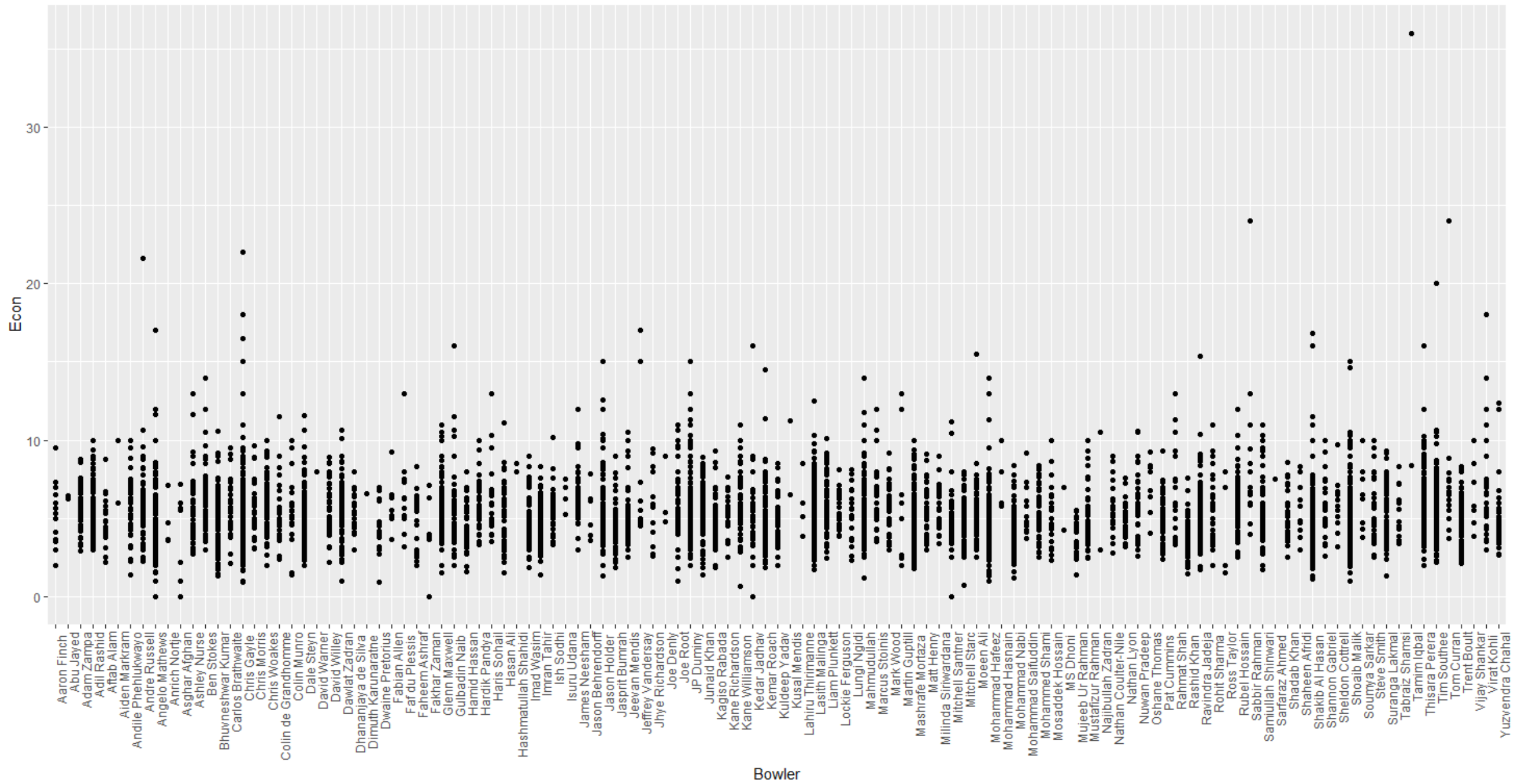
Since this code gives this……..

Since our dataset has almost all the players such as batsmen bowlers and also all-rounders we had to create another .csv file with only bowlers.

We did this……

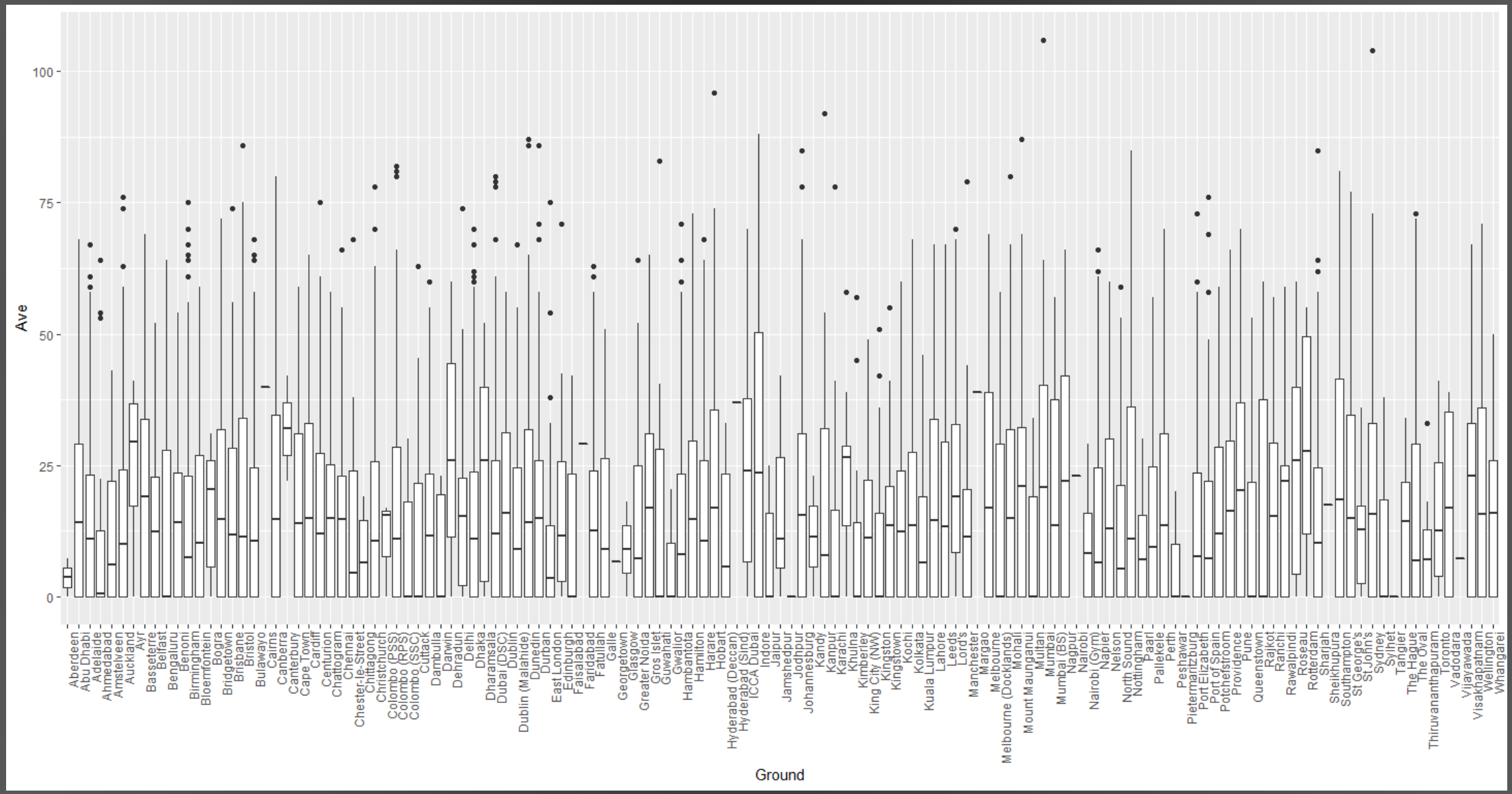
```
88  bowlers = df_sorted[apply(df_sorted[c(3)],1,function(z) any(z!=0)),]
89  write.csv(bowlers,file = 'only_bowlers.csv')
90  head(bowlers)
```

```
91
92  e<-ggplot(bowlers,aes(x=Bowler,y=Econ))
93  print(e+geom_point()+geom_smooth(method="loess",se=F)
94        +theme(axis.text.x = element_text(angle = 90, hjust = 1)))
95
```


Econ
30
20
10
0
Bowler
Aaron Finch
Abu Jayed
Adam Zampa
Adil Rashid
Aftab Alam
Aiden Markram
Andile Phehlukwayo
Andre Russell
Angelo Mathews
Anrich Nortje
Asghar Afghan
Ashley Nurse
Ben Stokes
Bhuvneshwar Kumar
Carlos Brathwaite
Chris Gayle
Chris Morris
Chris Woakes
Colin de Grandhomme
Colin Munro
Dale Steyn
David Warner
David Willey
Dawlat Zadran
Dhananjaya de Silva
Dimuth Karunaratne
Dwaine Pretorius
Fabian Allen
Faf du Plessis
Faheem Ashraf
Fakhar Zaman
Glenn Maxwell
Gulbadin Naib
Hamid Hassan
Hardik Pandya
Haris Sohail
Hasan Ali
Hashmatullah Shahidi
Imad Wasim
Imran Tahir
Ish Sodhi
Isuru Udana
James Neesham
Jason Behrendorff
Jason Holder
Jasprit Bumrah
Jeevan Mendis
Jeffrey Vandersay
Jhye Richardson
Joe Denly
Joe Root
JP Duminy
Junaid Khan
Kagiso Rabada
Kane Richardson
Kane Williamson
Kedar Jadhav
Kemar Roach
Kuldeep Yadav
Kusal Mendis
Lahiru Thirimanne
Lasith Malinga
Liam Plunkett
Lockie Ferguson
Lungi Ngidi
Mahmudullah
Marcus Stoinis
Mark Wood
Martin Guptill
Mashrafe Mortaza
Matt Henry
Milinda Siriwardana
Mitchell Santner
Mitchell Starc
Moeen Ali
Mohammad Hafeez
Mohammad Hasnain
Mohammad Nabi
Mohammad Saifuddin
Mohammed Shami
Mosaddek Hossain
MS Dhoni
Mujeeb Ur Rahman
Mustafizur Rahman
Najibullah Zadran
Nathan Coulter-Nile
Nathan Lyon
Nuwan Pradeep
Oshane Thomas
Pat Cummins
Rahmat Shah
Rashid Khan
Ravindra Jadeja
Rohit Sharma
Ross Taylor
Rubel Hossain
Sabbir Rahman
Samiullah Shinwari
Sarfaraz Ahmed
Shadab Khan
Shaheen Afridi
Shakib Al Hasan
Shannon Gabriel
Sheldon Cottrell
Shoaib Malik
Sourmya Sarkar
Steve Smith
Suranga Lakmal
Tabraiz Shamsi
Tamim Iqbal
Thisara Perera
Tim Southee
Tom Curran
Trent Boult
Vijay Shankar
Virat Kohli
Yuzvendra Chahal

7. Average vs Ground

```
96  g<-ggplot(bowlers,aes(x=Ground,y=Ave))
97  print(g+geom_boxplot()+geom_smooth(method="loess",se=F)
98        +theme(axis.text.x = element_text(angle = 90, hjust = 1)))
99
```



8. Bowler' 5 wickets vs Opposition

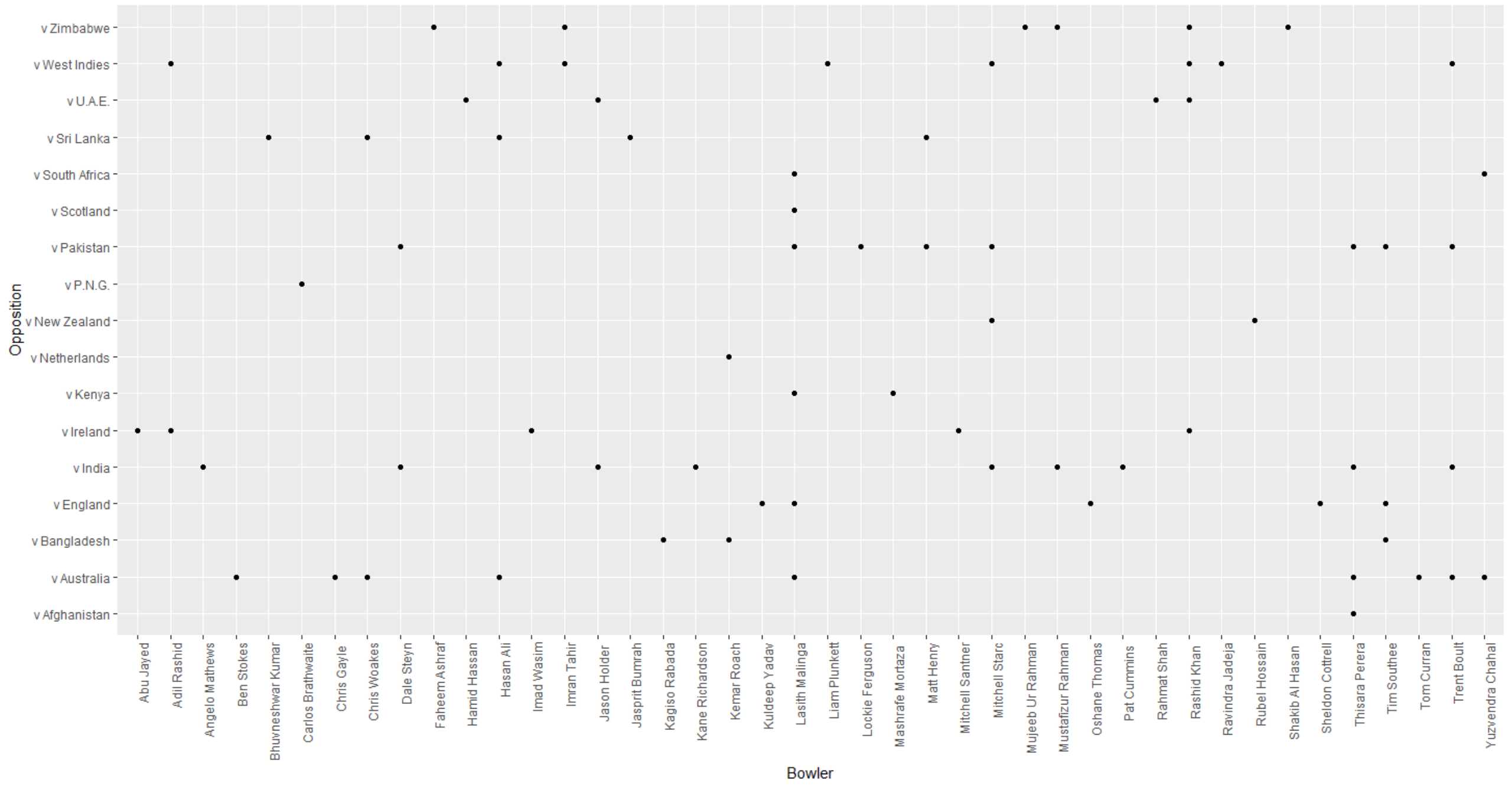
```
100  bowlers_5 = subset(bowlers,Wkts>=5)
101  write.csv(bowlers_5,file = 'bowlers_5_wkts.csv')
102  head(bowlers_5)
```

The 'only_bowlers.csv' stored in 'bowlers' has all the bowlers but we need the bowler's dataset with bowler's who have taken 5 or more 5 wickets. Hence the below code fulfills our need.

```
> head(bowlers_5)
       X.1     X Overs Mdns Runs Wkts Econ  Ave  SR    Opposition          Ground Start.Date   Match_ID         Bowler Player_ID
6863    90    90   5.1    0   21    5 4.06 4.20 6.2     v England        Gros Islet   2-Mar-19 ODI # 4103  Oshane Thomas    914567
6892 10369 10369   5.3    0   32    5 5.81 6.40 6.6      v U.A.E.      Kuala Lumpur   2-May-14 ODI # 3488    Rahmat Shah    533956
6899  3366  3366   5.4    0   14    5 2.47 2.80 6.8     v Ireland Dublin (Malahide)  18-Aug-16 ODI # 3767     Imad Wasim    227758
6911  8260  8260   5.5    0   26    6 4.45 4.33 5.8 v New Zealand            Dhaka  29-Oct-13 ODI # 3423  Rubel Hossain    300619
7002  1744  1744   6.0    0   20    6 3.33 3.33 6.0       v India     Colombo (RPS)  12-Sep-09 ODI # 2887 Angelo Mathews     49764
7025  2086  2086   6.0    1   24    6 4.00 4.00 6.0    v Zimbabwe      Bloemfontein   3-Oct-18 ODI # 4050    Imran Tahir     40618
```

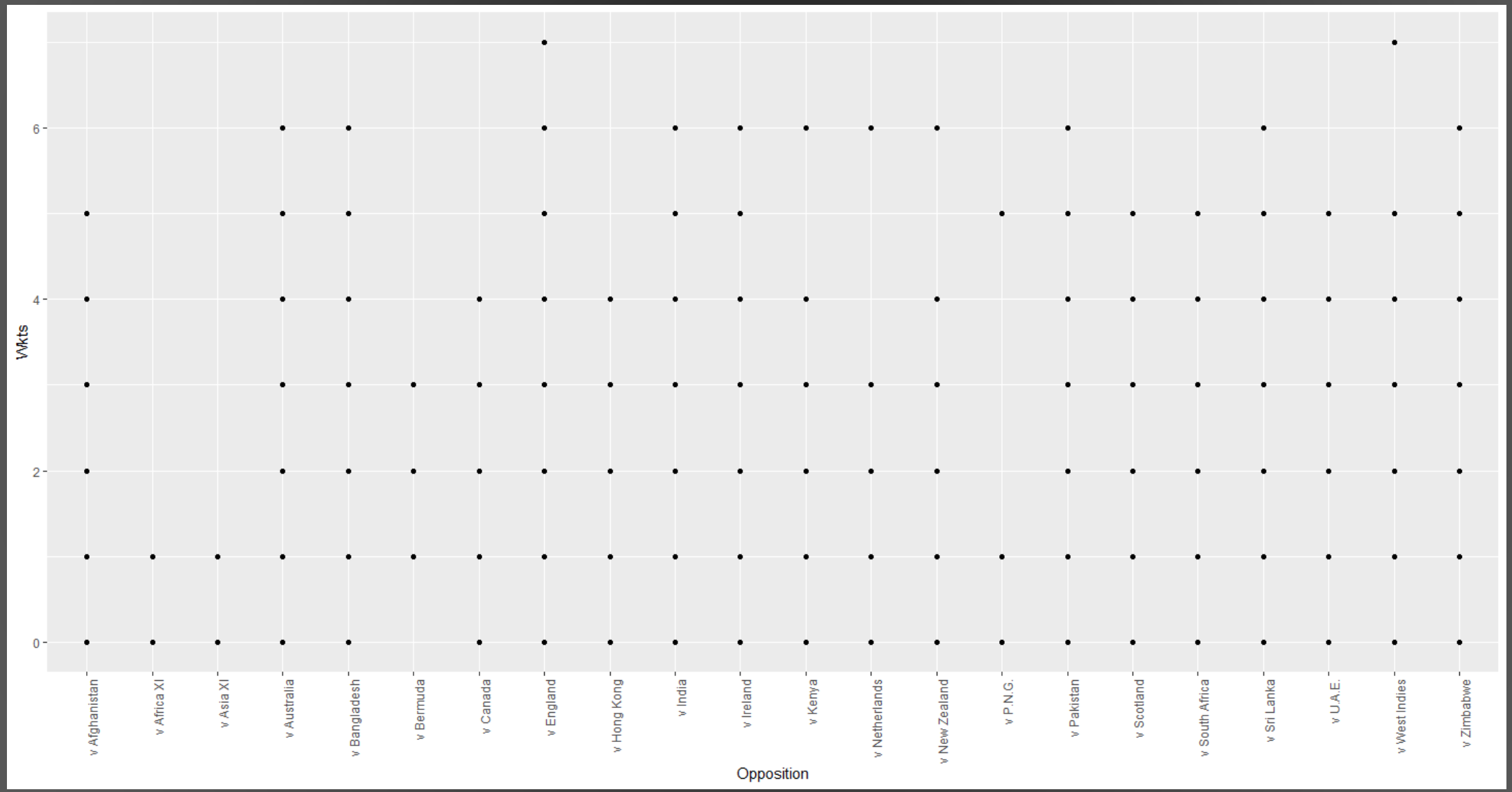
```
> tail(bowlers_5)
       X.1     X Overs Mdns Runs Wkts Econ   Ave SR Opposition     Ground Start.Date   Match_ID          Bowler Player_ID
10870  9639  9639    10    2   42    5  4.2  8.40 12 v Pakistan    Sharjah  28-Aug-12 ODI # 3299   Mitchell Starc    311592
10877  9660  9660    10    2   43    6  4.3  7.16 10    v India  Melbourne  18-Jan-15 ODI # 3582   Mitchell Starc    311592
10902  9824  9824    10    1   68    5  6.8 13.60 12    v India   Canberra  20-Jan-16 ODI # 3726  Kane Richardson    272262
10981 10170 10170    10    0   70    5  7.0 14.00 12    v India     Mohali  10-Mar-19 ODI # 4111     Pat Cummins    489889
11001 10530 10530    10    0   50    5  5.0 10.00 12 v Zimbabwe    Sharjah  16-Feb-18 ODI # 3977 Mujeeb Ur Rahman    974109
11058 10799 10799    10    1   45    5  4.5  9.00 12   v U.A.E. ICCA Dubai   4-Dec-14 ODI # 3562     Hamid Hassan    311427
```

```
103
104  h<-ggplot(bowlers_5,aes(x=Bowler,y=Opposition))
105  print(h+geom_point(method="loess",se=F)+theme(axis.text.x = element_text(angle = 90, hjust = 1)))
```


Opposition
v Zimbabwe
v West Indies
v U.A.E.
v Sri Lanka
v South Africa
v Scotland
v Pakistan
v P.N.G.
v New Zealand
v Netherlands
v Kenya
v Ireland
v India
v England
v Bangladesh
v Australia
v Afghanistan
Abu Jayed
Adil Rashid
Angelo Mathews
Ben Stokes
Bhuvneshwar Kumar
Carlos Brathwaite
Chris Gayle
Chris Woakes
Dale Steyn
Faheem Ashraf
Hamid Hassan
Hasan Ali
Imad Wasim
Imran Tahir
Jason Holder
Jasprit Bumrah
Kagiso Rabada
Kane Richardson
Kemar Roach
Kuldeep Yadav
Lasith Malinga
Liam Plunkett
Lockie Ferguson
Mashrafe Mortaza
Matt Henry
Mitchell Santner
Mitchell Starc
Mujeeb Ur Rahman
Mustafizur Rahman
Oshane Thomas
Pat Cummins
Rahmat Shah
Rashid Khan
Ravindra Jadeja
Rubel Hossain
Shakib Al Hasan
Sheldon Cottrell
Thisara Perera
Tim Southee
Tom Curran
Trent Boult
Yuzvendra Chahal
Bowler

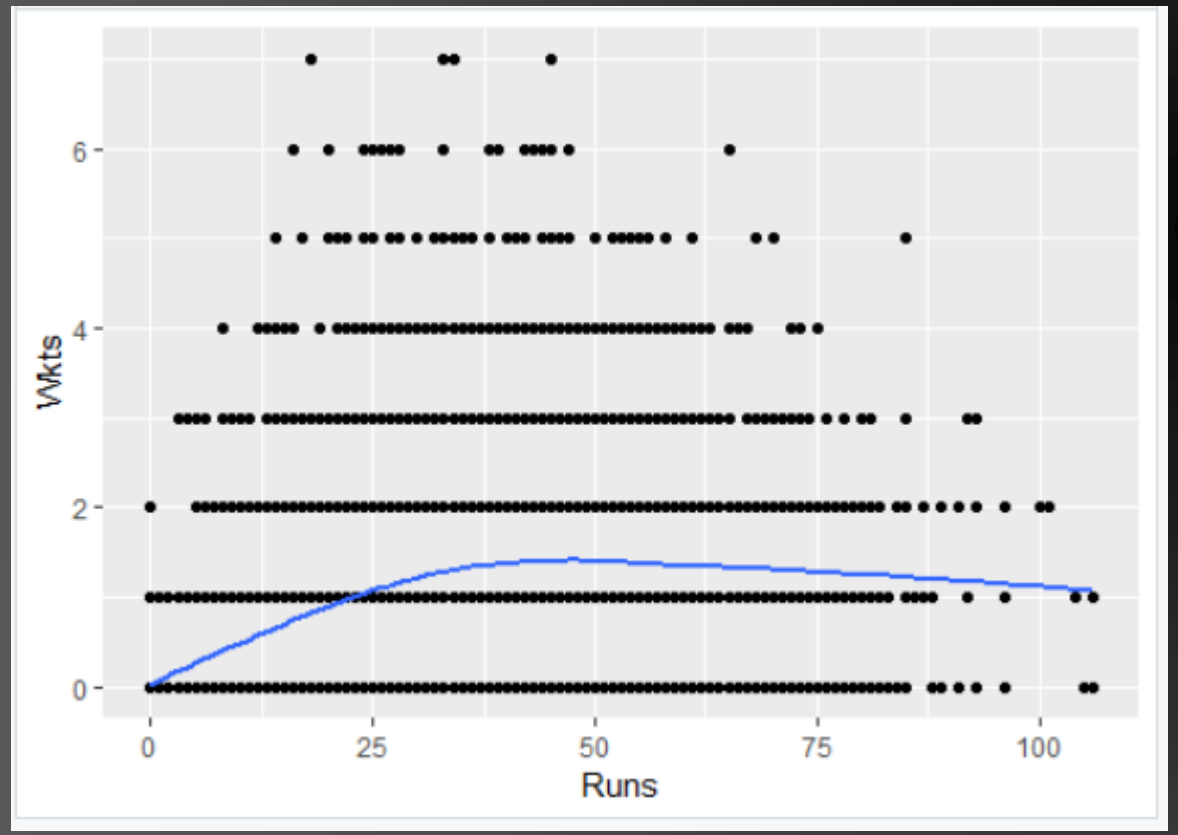
9. Opposition vs no. of wickets

```
107  i<-ggplot(bowlers,aes(x=Opposition,y=Wkts))
108  print(i+geom_point()+geom_smooth(method="loess",se=F)
109        +theme(axis.text.x = element_text(angle = 90, hjust = 1)))
110
```

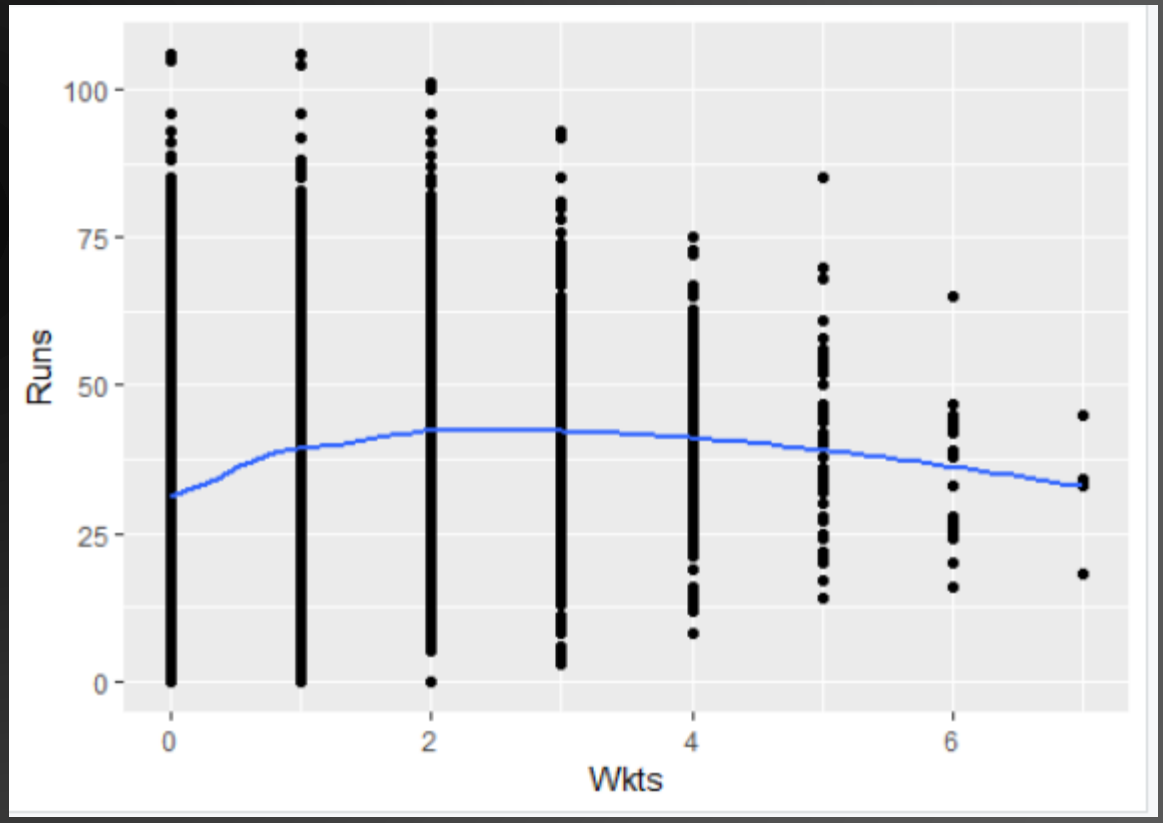


10. Runs vs Wickets

```
111  j<-ggplot(bowlers,aes(x=Runs,y=Wkts))
112  print(j+geom_point()+geom_smooth(method="loess",se=F))
113
```

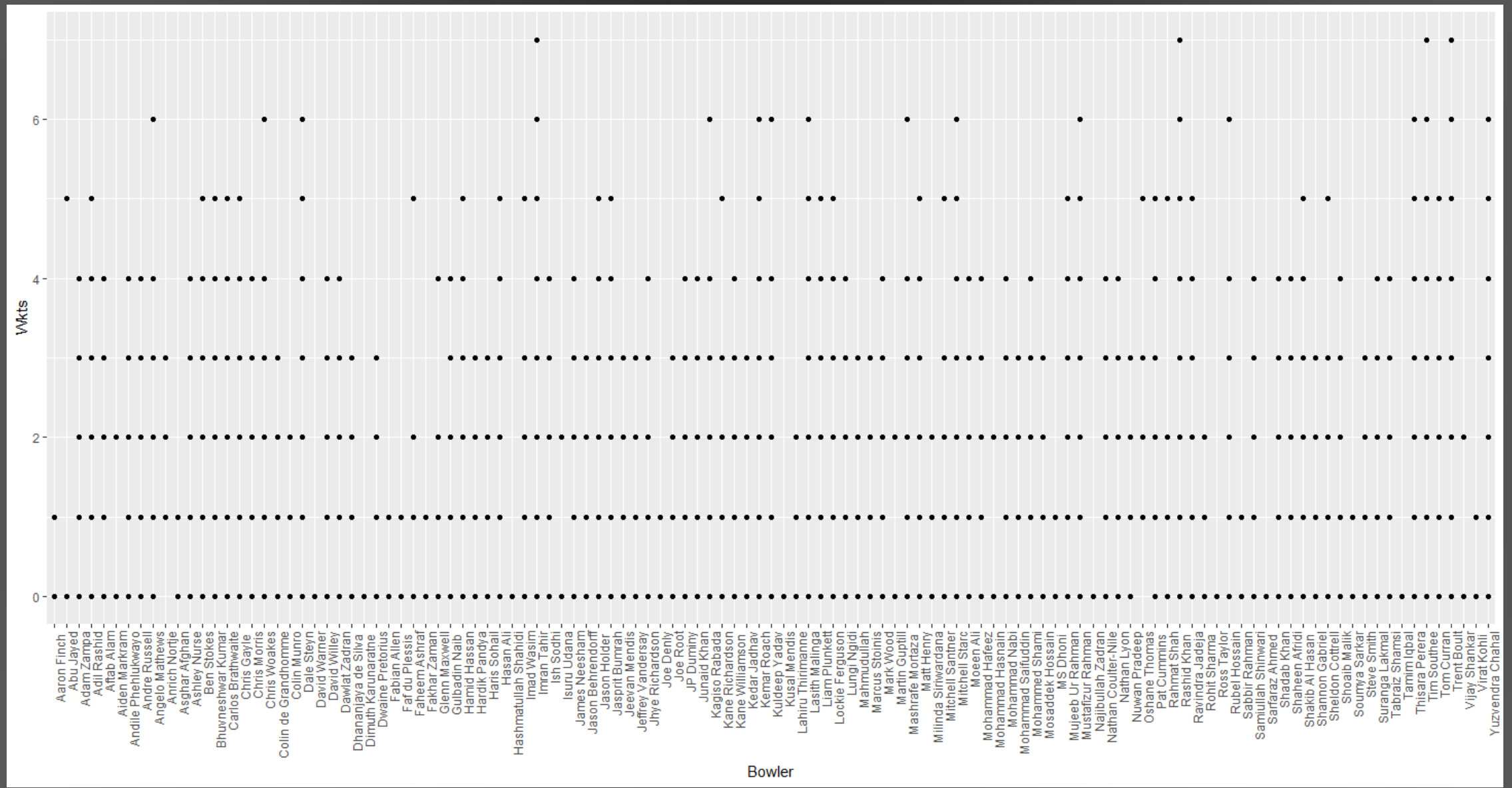



```
111  j<-ggplot(bowlers,aes(x=Wkts,y=Runs))
112  print(j+geom_point()+geom_smooth(method="loess",se=F))
113
```



11. Bowlers vs Wickets

```
114  k<-ggplot(bowlers,aes(x=Bowler,y=Wkts))
115  print(k+geom_point()+geom_smooth(method="loess",se=F)
116        +theme(axis.text.x = element_text(angle = 90, hjust = 1)))
117
```



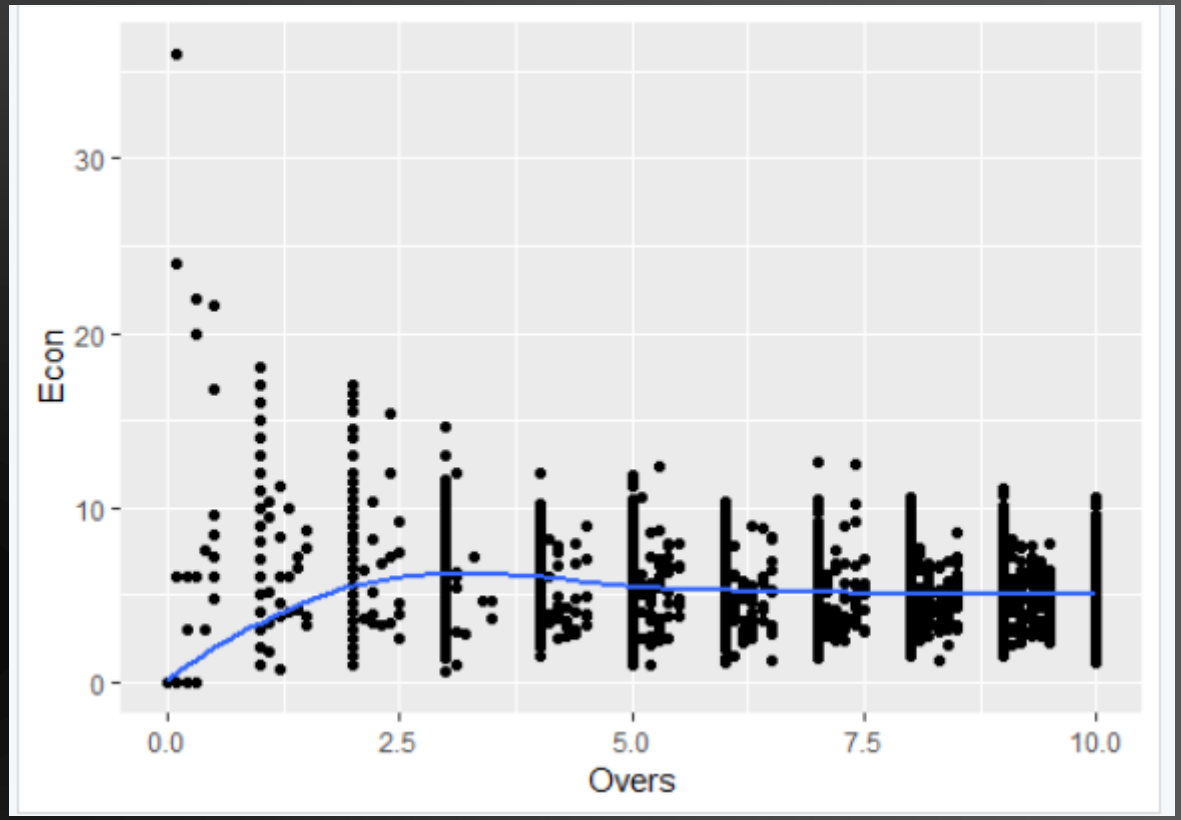
LINEAR REGRESSION
AND
HYPOTHESIS TESTING

LINEAR REGRESSION

```
118  #Linear Regression
119
120  print(cor(df_sorted$Overs, df_sorted$Econ)) #The model will be built for this dataset
121  linearMod<-lm(df_sorted$Overs~df_sorted$Econ, data=df_sorted)
122  print(linearMod)
123  print(summary(linearMod))
124
```

```
> print(cor(df_sorted$Overs, df_sorted$Econ)) #The model will be built for this dataset
[1] 0.6953931
> linearMod<-lm(df_sorted$Overs~df_sorted$Econ, data=df_sorted)
> print(linearMod)

Call:
lm(formula = df_sorted$Overs ~ df_sorted$Econ, data = df_sorted)

Coefficients:
   (Intercept)  df_sorted$Econ
        1.1324          0.9392

>
```

```
> print(summary(linearMod))

Call:
lm(formula = df_sorted$Overs ~ df_sorted$Econ, data = df_sorted)

Residuals:
    Min      1Q  Median      3Q     Max
-34.845  -1.132  -1.132   2.129   7.834

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.132375   0.038239   29.61   <2e-16 ***
df_sorted$Econ 0.939243   0.009206  102.02   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.965 on 11116 degrees of freedom
Multiple R-squared:  0.4836,    Adjusted R-squared:  0.4835
F-statistic: 1.041e+04 on 1 and 11116 DF,  p-value: < 2.2e-16

>
```




```
124
125  linear_regression_result<-ggplot(df_sorted,aes(x=Overs,y=Econ))
126  print(linear_regression_result+geom_point()+geom_smooth(method="loess",se=F))
127
```

Hypothesis Testing

```
128  #Hypothesis Testing
129  data1 <- df_sorted$Overs
130  normalize <- function(x) {
131    return ((x - min(x)) / (max(x) - min(x)))
132  }
133  data2<-normalize(data1)
134
135  var(data1,data2)  #variance
136  #[1] 1.701664
137
```

```
138  # var.test() function performs F-test between 2 normal populations
139  # with hypothesis that variancesof the 2 populations are equal.
140  # ... Since the p-value = 0.5242, which is much higher than 0.05,
141  # the hypothesis that the variances of x and y are equal is accepted.
142
143  var.test(data1,data2)
144
```

```
> var.test(data1,data2)

        F test to compare two variances

data:  data1 and data2
F = 100, num df = 11117, denom df = 11117, p-value < 2.2e-16
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  96.35029 103.78796
sample estimates:
ratio of variances
               100

>
```

```
145  mean(data1)    #[1] 3.776552
146  mean(data2)    #[1] 0.3776552
147
```

```
148 t.test(data2,mu=0.3776552,alternative="less")
149
```

```
> t.test(data2,mu=0.3776552,alternative="less")

        One Sample t-test

data:  data2
t = -1.1808e-05, df = 11117, p-value = 0.5
alternative hypothesis: true mean is less than 0.3776552
95 percent confidence interval:
      -Inf 0.3840907
sample estimates:
mean of x
0.3776552

>
```

```
149
150 t.test(data2,mu=0.3776552,alternative="greater")
```

```
> t.test(data2,mu=0.3776552,alternative="greater")

        One Sample t-test

data:  data2
t = -1.1808e-05, df = 11117, p-value = 0.5
alternative hypothesis: true mean is greater than 0.3776552
95 percent confidence interval:
 0.3712196       Inf
sample estimates:
mean of x
0.3776552

>
```

For regression case:

Null hypothesis: The beta coefficients are equal to zero.
Alternate hypothesis: The beta coefficients are not equal to zero.

Reject the null hypothesis as the p-value is very small.
So, the coefficients are significant.

Open to any Questions…


THANK YOU